A Stable Linked Structure Flooding for Mobile Ad Hoc Networks with Fault Recovery

Tom Leclerc, Laurent Ciarletta, André Schaff

A Stable Linked Structure Flooding for Mobile Ad Hoc Networks with Fault Recovery

Tom Leclerc, Laurent Ciarletta, and André Schaff

LORIA - Campus Scientifique - BP 239 - 54506 Vandœuvre-les-Nancy Cedex
{tom.leclerc,laurent.ciarletta,andre.schaff}@loria.fr

Abstract. Efficient message dissemination in ad hoc networks can be fostered by exploiting stable (sub-)structures. By efficient we mean low network resource usage regarding reachability. In this paper we build a hierarchical protocol. We first create single-hop clusters among stable-connected devices. On top of those clusters, we further determine inter-cluster relays (ICR), finally providing an overall stable-connected structure. Our proposed stable linked structure flooding (SLSF) protocol efficiently disseminates data among stable nodes. Additional fault recovery mechanisms are employed to compensate for local intermittent node failures if needed. The experiments show that our approach increases flooding performances with a low bandwidth usage. Furthermore SLSF remains very efficient with or without the fault recovery mechanism that provides robustness.

1 Introduction

Mobile ad hoc networks are composed of a collection of devices that communicate with each other over a wireless medium [1]. Such a network can be formed spontaneously whenever devices are in transmission range. Potential applications of such networks can be found in traffic scenarios, environmental observations, ubiquitous Internet access, and in search and rescue scenarios as described in detail in [2]. However, since joining and leaving of nodes occurs dynamically, the network topology changes frequently. Our solution starts by building local groups of one-hop stable-connected devices in a self-organizing manner. Moreover, the approach we introduced in [3] aims at discovering stable connections between groups, thus creating bigger stable-linked network structures. We exploit the stable-linked structures within the network topology to streamline information exchange and to minimize the overhead. The local one-hop groups are built using the NLWCA clustering protocol [4]. As in WCPD [5] specific beacon formats are used to detect nearby stable-connected clusters. Furthermore, to create bigger stable-connected structures we present Inter-Cluster Relays (ICR). Finally, to add robustness, a fault recovery protocol is employed to compensate for local intermittent node failures. Fault recovery can be selectively enabled/disabled on per-packet basis.

The remainder of this paper is organized as follows. The next Section 2 sets the context and presents the related work for this paper. Section 3 describes the

building blocks of our SLSF protocol. In Section 4 we evaluate our approach and present the simulation results. We conclude our paper and present the future work in section 5.

2 Context & Related Work

2.1 NLWCA & WCPD

The Weighted Cluster-based Path Discovery protocol (WCPD) is designed to take advantage of the cluster topology built by the Node and Link Weighted Clustering Algorithm (NLWCA) in order to provide path discovery and broadcast mechanisms in mobile ad hoc networks.

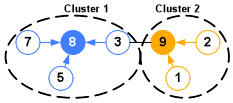

Fig. 1. Example of two clusters built by NLWCA.

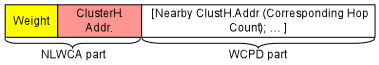

Fig. 2. NLWCA+WCPD Beacon.

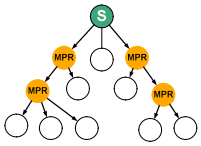

Fig. 3. OLSR topology from one source.

NLWCA organizes ad hoc networks in one-hop clusters (Figure 1) by using only information available locally. Each device elects exactly one device as its clusterhead (CH), i.e. the neighbor with the highest weight. So far, a topological chain can be formed by so called sub-head nodes. A sub-head is a node that elects a neighbor node as CH but at the same time is elected as CH by some other one-hop neighbor nodes. However, sub-heads can lead to more than three hops between a source CH and its nearby CHs which could lead to a complex communication protocol. To obtain strict one-hop clusters, thus simplifying the protocol, a rule was added to the original NLWCA algorithm: a node that already elected a foreign node as CH is not eligible to be elected by another node as CH. From a graph theory point of view, one-hop clusters form a dominating set.

The main goal of NLWCA is to avoid superfluous re-organization of the clusters, particularly when clusters cross each other. To achieve this, NLWCA assigns weights to the links between the own node and the network neighbor nodes. This weight is used to keep track of the connection stability to the one-hop network neighbors. When a link weight reaches a given stability threshold it is considered stable and the device is called stable neighbor device. The CH is elected only from the set of stable neighbors which avoids the re-organization of the topology when two clusters are crossing for a short period of time.

WCPD, on top of NLWCA, discovers nearby stable-connected clusters in a pro-active fashion. For the nearby CHs discovery algorithm, WCPD uses the beacon to detect devices in communication range. NLWCA and WCPD combined provide to each node, through the beacon (Figure 2), following information about

each stable one-hop neighbor: its weight, its CH ID, the ID set of discovered CHs and their respective path length.

The WCPD broadcasting algorithm is simple: the broadcast source node sends the message to the CH, which stores the ID of the message and broadcasts it to the one-hop neighborhood. After that, it sends it to all nearby CHs by multi-hop unicast. The inter-cluster destination nodes repeat the procedure except that the message source clusters are omitted from further forwarding. Additionally, the information about the ID of the broadcast messages and their sources is stored for a given period of time to avoid superfluous re-sending of the message.

2.2 OLSR

The Optimized Link State Routing Protocol (OLSR) [6] is a well known routing protocol designed for ad hoc networks. It is a proactive protocol; hence it periodically exchanges topology information with other nodes of the network. One-hop neighborhood and two-hop neighborhood are discovered using Hello Messages (similar to beacons). The multipoint relay (MPR) nodes are calculated by selecting the smallest one-hop neighborhood set needed to reach every two-hop neighbor node. The topology control information is only forwarded by the nodes which are selected as MPR. Every node then has a routing table containing the shortest path to every node of the network. OLSR enables optimized flooding of the network by building a tree-like topology for every node from a source (Figure 3). Therefore, MPR selection constructs an optimal connected dominating set [7]. For this reason OLSR is our benchmark reference in Section 4.

2.3 Related Work

In ad hoc networks forwarding strategies should be employed to avoid broadcast storms (i.e. a message forwarded by all the nodes in the network). As depicted in [8], broadcast storms can be counter-measured using several schemes i.e. probabilistic, counter-based, distance-based, location-based and cluster-based. We use the latter scheme, cluster-based, since it is the only one based on network topology information. The cluster architecture we use solely relies on locally available information whereas [8] proposes a clustering technique where the CH is elected after an explicit message exchange among the neighbors.

The Zone Routing Protocol (ZRP)[9] combines proactive routing inside a zone using bordercast and on-demand routing outside. A node, sending a message, checks if the destination is in its zone, if not it bordercasts the message to its gateway nodes. Those nodes repeat the same process until the destination is reached. As every zone is centered on the current node the ZRP dissemination results in plain bordercasting a message ahead its destination. To route inside a zone, ZRP needs k-hop information (k>1), which results in scalability issues similar to OLSR.

In [10], the authors construct elected clusters based on beacon information which provides the number of neighbors and their stability represented by the number of beacons received. This approach is similar to NLWCA but does not

rely on both a link weight and a node weight. Thus NLWCA has more flexibility in terms of cluster selection. Another similarity to our approach is the forwarding node selection (named gateway selection). The main differences is that our approach requires no message exchange except the payload broadcast itself for gateway selection. The comparison will not be further analyzed since to our knowledge the gateway selection process is insufficiently described.

Many ad hoc protocols use the selection of forwarding nodes to reduce redundant messages. In [11] and [12] broadcast relay gateways are selected with 2-hop knowledge. However we chose the already described OLSR protocol because of its popularity in ad hoc networks and Mesh networks, and also because it is the only one proved to optimize coverage of 2-hop nodes through MPRs [13]. MPRs build optimal connected dominating sets [7]. We take advantage of this property to build paths among the (not-connected) dominating sets built by the NLWCA clustering algorithm. Many other approaches that construct distributed connected dominating sets exist [14, 15]. However, our goal is not to create connected dominating sets, but to disseminate information over the **stable** structure built by NLWCA, using ICRs between the CHs to optimize nearby-cluster-paths. Thus, as a result of our structure, we have connected dominating sets, but only between a CH and its nearby CHs.

3 SLSF

NLWCA and WCPD provide a stable-connected cluster architecture, however the broadcasting algorithm of WCPD needs many improvement on reachability performances compared to OLSR which performs very well on reachability but lacks in scalability and uses a lot of bandwidth [16]. Our SLSF (Stable Linked Structure Flooding) protocol replaces the inefficient broadcasting mechanism of WCPD with the ICR mechanism. SLSF combines the advantages of all the protocols NLWCA, WCPD and OLSR: scalability, stability, reachability, while keeping the drawbacks low (i.e. the bandwidth usage). SLSF forms a first level of hierarchy with a dominating set using NLWCA. Considering the dominant nodes of the underlying level (NLWCA), it forms an optimal connected dominating set with the ICR mechanism. The first level reduces the network to its dominant nodes and the second level insures shortest-path connectivity and minimal relay nodes among dominant nodes of the first level.

3.1 SLSF - Inter-Cluster Relay

Multiple paths to reach a given CH requires choosing one path prior to another. We use a next-hop selection inspired by the MultiPoint Relay (MPR) mechanism of OLSR to select the forwarding neighbors. We name Inter-Cluster-Relays (ICR) the nodes selected as next-hop. The goal of ICR is to reach all nearby CHs with the minimal set of 1-hop neighbors while optimizing the hop-count. The ICR nodes are calculated by selecting the smallest one-hop neighborhood set (directly connected nodes) needed to reach every nearby CH. ICR selection

remains straightforward because the possible inter-cluster configurations are restricted by the underlying one-hop cluster topology (examples on Figure 6).

SLSF, on top of NLWCA, discovers the nearby clusters (similar to WCPD) by reading the neighbor beacons. The improvement and novelty relies on the ICR selection which avoids superfluous network communication overhead without any additional message exchange. SLSF keeps the last beacon of every one-hop neighbor in cache. Hence every node has the following information locally available about each stable 1-hop neighbor: its weigth, its CH ID, the ID set of discovered CHs and their respective path length. ICR selection occurs as follows:

i. Select as ICR, neighbors that are the only one reaching a given nearby CH.
ii. Remove the now covered clusters from the list.
iii. Remove for every neighbor from the announced CH-list the entries with a worse hop count than the best one (i.e. keep only shortest path entries).
    1. Calculate the cluster reachability for every one-hop neighbor (i.e. number of foreign CHs the neighbor announces in its beacon).
    2. Select the neighbor with the best reachability.
    3. Else if equivalent: select the neighbor with the highest weight.
    4. Else if equivalent: select the node with the biggest IP address.
    5. Remove the now covered clusters from the list.
    6. While there is a not-covered CH, go back to 1.

3-hop Inter-cluster case. NLWCA builds one-hop clusters, thus it permits up to three hops (two slave nodes) between CHs. ICR selection with two hops (one slave node) between CHs (Figure 6b) is straight selection by the CH, however an additional hop (Figure 6d) requires additional attention.

A further hop involves an additional forward of the message to reach the nearby CH. For example on Figure 6d, the source CH2 designates a node as ICR (here the blue slave neighbor of CH2). The designated ICR has to make a choice between one of the two (orange) slaves of CH1. This choice is computed by using ICR selection using the list containing only 1 hop distant(from the blue slave: here the orange CH) clusterheads.

When to select ICR nodes? ICR selection is done based on events. Every time a change in the stable neighborhood that influences the ICR calculation occurs, the ICR selection is re-calculated. Thus broadcasting or forwarding a message using ICRs is immediate: replace the ICR set in the message with the one locally pre-calculated. Further detail on broadcasting in SLSF in section 3.2.

ICR: The big picture. To highlight the gain of ICR selection, Figure 4 shows an example with 5 clusters where the message source CH S sends a broadcast. The broadcast of S will have the format shown on figure 5.

On reception of this broadcast only nodes 1, 2 and 3 will forward the message, while the other neighboring nodes process the message silently. Note that node 1 selects 4 and 5 as ICR according to section 3.1 "3-hop Inter-cluster case".

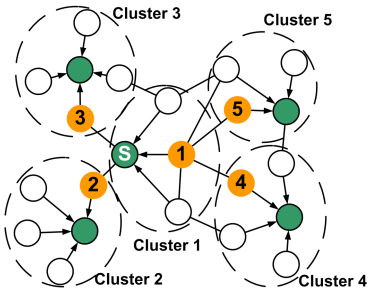

Fig. 4. ICR selection with 5 clusters.

Fig. 5. Format of a broadcast message with payload.

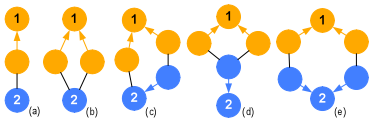

Fig. 6. Inter-cluster configuration examples where 1 and 2 are clusterheads.

We see that ICR selection reduces a lot the number of forwarding nodes. As an example, on Figure 4 there are 23 nodes in the network and only 10 nodes (including the CHs) are emitting to reach all the nodes in the network. Every CH will emit the message once in order for their slave to receive it and if necessary include their local ICR selection for further forwarding in the network (see section 3.2). In comparison, there would be 15 nodes forwarding the message using OLSR. This is due to OLSR using only 2-hop information while SLSF uses 1-hop cluster information which represent information from up to 3-hops away. While 3-hop knowledge usually increases the amount of information to collect using clusters reduces drastically the nodes to keep track of for ICR selection.

3.2 SLSF - Broadcast

At this point, every communication occurs between one cluster and its nearby clusters. To enable communication with foreign clusters, we propose a simple broadcast mechanism. A more sophisticated foreign cluster-broadcast mechanism would be out of scope for this paper.

Our broadcast mechanism is simple now that we only need to deal at cluster level. A node willing to broadcast a message through the network will, unless it is its own CH, send it to its CH. The original message contains the source address, a corresponding sequence number and of course the payload data. The CH puts its own address as last crossed CH and adds a corresponding sequence number to the message. Finally it forwards it to all its nearby CHs using the ICR mechanism. On reception the nearby CHs replace the last crossed CH address with their own and replace the corresponding sequence number. The ICR (excluding from selection the cluster the message came from) set is also updated. To avoid superfluous re-sending of the message SLSF stores information about the ID of messages and their sources for a given period of time.

As an example, on Figure 7 the node 1 sends a message passing (as 1 is a slave) the message to CH9, which then uses the ICR mechanism for the dissemination. Note that node 7 is in the "3-hop inter-cluster case" (section 3.1). The message reaches all the nodes in the network following the path depicted on Figure 8.

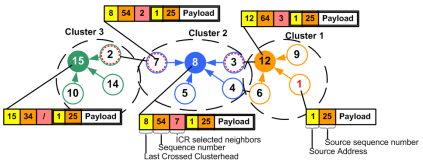

Fig. 7. Foreign-cluster broadcast - Format of a message sent from node 1.

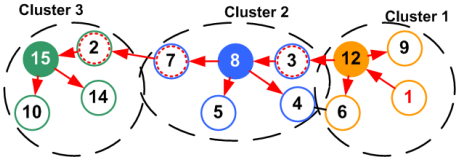

Fig. 8. Foreign-cluster broadcast - Path of a message sent from node 1.

3.3 SLSF - Fault-recovery

The goal of fault-recovery is to be able to transmit the message even if the ICR path fails without having the source to re-emit the message. To do so we propose two mechanisms, the first is the acknowledgement mechanism, enabling broadcasts to be acknowledged by nearby CHs and the second is the delayed transmission detecting transmission errors and handling them.

Acknowledgment mechanism The following acknowledgement mechanism has two advantages. The first, for which it was actually designed, is permitting a delayed transmission to compensate local intermittent node failures. The second is classic acknowledgment of messages but only between adjacent CHs as opposed to acknowledgements from one end of the network to the other. The classic cluster-to-cluster acknowledgement is a consequence of the initial design. Further consideration of end-to-end acknowledgements would be out of scope for this paper but is an open interest for future work.

Our acknowledgement mechanism works using sequence numbers. Every node puts inside its beacon sequence numbers to acknowledge messages. However, only CHs acknowledge the messages while inter-cluster nodes forward merely the acknowledgement information coming from nearby CHs to their neighborhood. As consequence, beacons of CHs have a different format than inter-cluster node beacons. Following is an example illustrating how the acknowledgement mechanism works. Using Figure 1 as reference, we suppose node 9 sends a broadcast with sequence number N9. Nodes 1, 2 and 3 receive the message. Only node 3 will forward it to its neighborhood as it is designated ICR, since it is the only inter-cluster node. Node 1 and 2 process silently the message without forwarding it. CH8 receives the message from node 3, processes the messages, and puts in its beacon the acknowledged sequence number N9 for CH9. Beacon of CH8 contains now its weight, its CH-address (here its own), its nearby CHs with the corresponding hop count and the acknowledged sequence number. So, the SLSF beacon (Figure 9a) is only extended with one sequence number (Figure 9b).

Node 3 reads the CH8 beacon (Figure 9b) and includes in its beacon the new sequence number acknowledged by CH8 for the message source CH9. In order to reduce the beacon size for inter-cluster nodes, we compact acknowledgment information into the basic SLSF beacon by just adding sequence numbers in the right order and place. If we consider the basic SLSF beacon (Figure 10a) for the slave node 3 we integrate additionally inside the beacon the information that

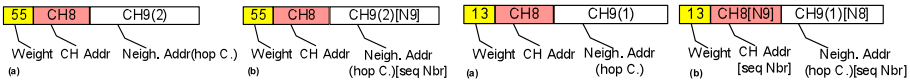

Fig. 9. CH8 Beacon: (a)SLSF, (b)SLSF with fault recovery.

Fig. 10. Node 3 Beacon: (a)SLSF, (b)SLSF with fault recovery.

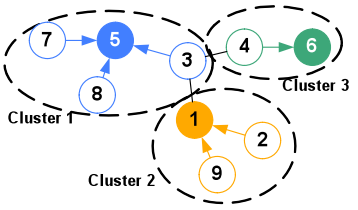

Fig. 11. Three clusters example.

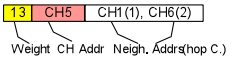

Fig. 12. Node 3 SLSF beacon.

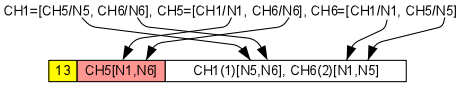

Fig. 13. Node 3 SLSF beacon with fault recovery .

CH8 acknowledges sequence number N9 for CH9 and that CH9 acknowledges N8 for CH8 (Figure 10b). Note the inversion of the sequence numbers compared to the beacon of CH 8.

To really point out this inversion an example where three nearby clusters are inter-connected is necessary (Figure 11).In this example, node 3 is connected to three clusters. As a matter of fact, node 3 will forward any message exchange between those clusters. The beacon of node 3 has to contain all the acknowledgement information for the three CHs. The basic SLSF beacon of node 3 of Figure 11 is shown in Figure 12.

Node 3's beacon contains its own clusterhead, CH5, and two nearby CHs, CH1 and CH6, on 1 and 2 hop distance respectively. Now we consider we want the following information into that beacon, like for instance after some broadcasts were sent (Nx corresponds to the acknowledged sequence number):

CH1 acknowledges N5 for CH5 and N6 for CH 6 $\Leftrightarrow$ CH1=[CH5/N5, CH6/N6]
CH5 acknowledges N1 for CH1 and N6 for CH 6 $\Leftrightarrow$ CH5=[CH1/N1, CH6/N6]
CH6 acknowledges N5 for CH5 and N1 for CH 1 $\Leftrightarrow$ CH6=[CH5/N5, CH1/N1]

The original SLSF beacon already contains the three CH-addresses (typically the IP addresses of the CHs); the own CH-address and two nearby CH-addresses. First sort all the acknowledgements CH-addresses in ascending order. For example CH6=[CH5/N5, CH1/N1] is sorted as CH6=[CH1/N1, CH5/N5]. For each CH-address in the beacon attach the corresponding sequence numbers and omit the superfluous CH-addresses inside the list, as shown on Figure 13. Doing so, putting in the right order the right sequence numbers, results in a new beacon with all the needed information just by adding the needed sequence numbers without additional addresses.

The constructed beacon (Figure 13) is received by the neighbors. For instance CH1 can reads in the beacon [N5,N6] and by ordering the announced CHs (CH1,

CH5, CH6) and omitting itself (CH5, CH6) it can read that CH5 acknowledges N5 and CH6 acknowledges N6.

The presented acknowledgement mechanism permits acknowledgements between a cluster and its nearby clusters. While the acknowledgements of the CHs using beacons are straightforward, the inter-cluster beacons of the nodes are constructed using inversion and re-ordering to avoid redundant information inside the beacon and still enabling the source cluster to distinguish which cluster acknowledges which messages without any other message exchange than beacons.

Delayed transmission mechanism The ICR selection enables the communication among nearby clusters in an optimized way. To keep the inter-cluster communication reliable in case the selected ICR path failed, we introduce a delayed transmission mechanism. Thus, we have an immediate communication path formed by the ICRs and we keep a backup, although delayed, path in case of failures.

Delayed transmission occurs as follows: If a broadcasted message is emitted from a source, all neighbor nodes NOT selected as ICR in the message will keep the message in cache and only in case of a failure transmit the message. While the ICR nodes forward the messages, the nodes pending for delayed transmission observe the neighboring beacons for acknowledgements of nearby CHs. On reception of a pending acknowledgement, the delayed transmission is aborted. If the pending time for delayed transmission times out, the message is broadcasted to all neighbors with the retransmitted flag set. Nodes that already received the message will discard it silently while others forward it immediately. The message arrives at destination in case of failure without re-emission of the message from the source. Thus, having only nodes that did not try forwarding the message yet, effectively (re)transmitting the message.

Following discusses the delay chosen for timeout to occur. If we consider the beacon-interval as bI, the number of nodes the message already passed through as hopCount and 4 the maximal number of hops for a message to go back and forth from the first slave to a cluster on (maximal) 2-hop distance. Then the transmission delay td is calculated as follows: $td = (\frac{4}{hopCount}) \times bI$. If the beacon-interval is 1 second and the hopcount is 1 then the transmission delay is set to 4 seconds. If the hopcount is 2 then the transmission delay is set to 2 seconds.

4 Simulation & Results

To evaluate the performances of our SLSF protocol, we implemented the three protocols (OLSR, NLWCA/WCPD and SLSF) on top of the JANE simulator [17] and performed several simulations. For those experiments we used the Restricted Random Way Point mobility model [18], whereby the devices move along defined streets on the map of Luxembourg City for 1000 seconds. For each device the speed was randomly varied between [0.5;1.5] units/s with a transmission range of 25 units. For each experiment 10 different random distribution seeds were used in order to feature results from different topologies and movement setups.

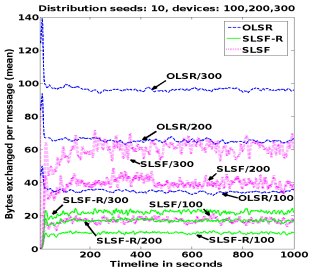

Fig. 14. Bandwidth used in order to build the topology for 100, 200 and 300 nodes

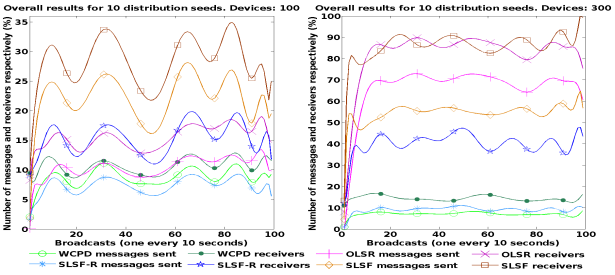

Fig. 15. Overall number of sent messages and node receivers for 100 and 300 nodes. (smoothed with a polynomial equation of the 16th grade for visibility sake).

For the used mobile environment where nodes move with low (walking) speeds between 1.8 and 5.4 km/h the NLWCA link-stability threshold is set on 2 [4].

Simulations were done to determine the bandwidth used by the protocols in order to build the topologies and the information dissemination performance of broadcasting on top of the different topologies. Then we compared the efficiency of the protocols and finally made a static evaluation to compare information dissemination solely on MPR and ICR performances.

OLSR exchanges the sets of one-hop neighbor nodes with every node in communication range. Similar to OLSR, SLSF exchanges the list of the discovered nearby-CHs with the one-hop neighbor nodes. For our experiments we distinguished two different SLSF configurations. the first is SLSF without fault recovery referred as SLSF-R (minus fault recovery), thus only ICR selection with basic SLSF beacon (same format as the WCPD beacon, thus same bandwidth usage). The second is the full SLSF protocol with ICR selection and fault-recovery mechanism with added sequence numbers in the beacon. To find out the network load produced during this phase, the size of the exchanged data sets were tracked every second of the simulation: for OLSR the size of the one-hop neighbor sets, for SLSF-R and WCPD the size of the discovered CHs and for SLSF additionally to the discovered CHs the sequence numbers.

In order to monitor the information dissemination performance (Reachability) a node was chosen to broadcast a message every 10 seconds during different simulation runs. The number of sent messages (i.e. broadcasts and unicasts) during the dissemination and the number of reached network nodes were tracked.

As shown in Figure 14, OLSR uses a higher bandwidth in both sparser (100 nodes) and denser networks (300 nodes). This was expected since OLSR is exchanging the set of one-hop neighbors, while SLSF only exchanges the set of locally discovered nearby CHs which is a fractional amount of the total number of nodes. SLSF-R uses exactly the same bandwidth (80% less than OLSR) as WCPD since they share the same beacon structure. SLSF uses more bandwidth than SLSF-R, as sequence numbers were added to the beacon, but still uses about 40% less bandwidth than OLSR.

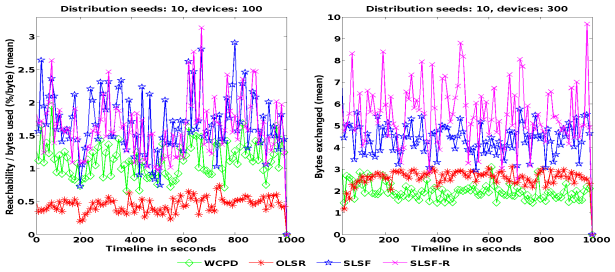

Fig. 16. Efficiency of Bandwidth usage for 100 and 300 nodes

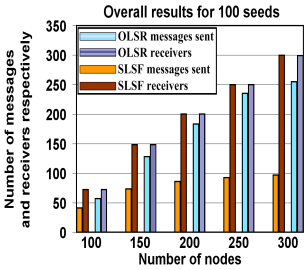

Fig. 17. Static scenario with 100 to 300 nodes

The dissemination performance results (Figure 15, NOTE: periodicity in curves is induced by the smoothing) show that SLSF performs the best for all densities. For 300 nodes SLSF performs only slightly better than OLSR, but uses on average 10 to 15% less forwarders with 40% less bandwidth usage. WCPD performs the worst and uses accordingly lesser forwarding nodes. Whereas SLSF-R uses approximately the same amount of forwarders than WCPD, it reaches from 10% to 20% more nodes. This is the pure gain of ICR selection which optimizes the forwarding nodes.

Subsequently we calculated a "quality-cost" ratio extracted from the results of Figures 14 and 15. We calculated the percentage of nodes reached, divided by the bandwidth used. We see in Figure 16 that SLSF and SLSF-R are in average two to three times more efficient than OLSR. The poor performances of WCPD highlight the need for improvement that SLSF brings.

SLSF relies on stable structures built by NLWCA: only nodes considered as stable will receive the message. So finally, to compare the performances on equal levels, we experimented OLSR and SLSF in a static scenario where all nodes are considered stable connected. The experiments where done on a 300x300 units surface with 100 to 300 nodes randomly positioned using 100 different topology seeds. Again, the number of forwarding and receiving nodes using MPR and ICR selection where tracked. The results on Figure 17 show that SLSF outperforms OLSR in terms of ratio of forwarding nodes over receiving nodes. With increasing density on average with OLSR about 85% of the receivers are also forwarders, whereas in SLSF this amount decreases from 60% towards 30%.

5 Conclusion & Future Work

This paper proposes SLSF, a flooding protocol which selects Inter-Cluster Relays to optimize the communication among the stable-connected cluster architecture. To deal with intermittent message loss we added a fault recovery mechanism that can be selectively enabled on a per-packet basis.

The goal of the ICR (Inter-Cluster Relay) selection is to reach all nearby clusterheads with the minimal set of 1-hop neighbors while optimizing the hop-count. ICR selection on top of the stable-cluster architecture reduces substan-

tially the number of forwarding nodes. Generally, SLSF performs well in high density networks while keeping the used bandwidth very low.

Currently we consider using SLSF as basis for Zerconf [19] (a service discovery protocol) in simulation and real world experiments.

As future work we plan to evaluate the performances using various mobility models and topology settings and also assess the results, in the context of the French National Research Project SARAH, deploying a large scale ad hoc network inside a museum.

References

1. Dousse, O., Thiran, P., Hasler, M.: Connectivity in ad-hoc and hybrid networks. In: IEEE Infocom 2002. (2002) 1079–1088
2. Santi, P.: Topology control in wireless ad hoc and sensor networks. ACM Comput. Surv. **37**(2) (2005)
3. Leclerc, T., Ciarletta, L., Schaff, A.: SLSF: stable linked structure flooding for mobile ad hoc networks. In: IEEE ISWPC, Palazzo Ducale, Modena, Italy (2010)
4. Andronache, A., Rothkugel, S.: Nlwca node and link weighted clustering algorithm for backbone-assisted mobile ad hoc networks. In: ICN '08, IEEE
5. Andronache, A., Rothkugel, S.: Hytrace backbone-assisted path discovery in hybrid networks. In: CTRQ '08, IEEE Computer Society (2008) 34–40
6. : Optimized link state routing protocol (olsr), rfc3626 (2003)
7. Adjih, C., Jacquet, P., Viennot, L.: Computing connected dominated sets with multipoint relays. Research Report RR-4597, INRIA (2002)
8. Ni, S.Y., Tseng, Y.C., Chen, Y.S., Sheu, J.P.: The broadcast storm problem in a mobile ad hoc network. In: MobiCom '99, NY, USA, ACM Press (1999)
9. Haas, Z.J., Pearlman, M.R., Samar, P.: The zone routing protocol (zrp) for ad hoc networks. Technical report (July 2002)
10. Foroozan, F., Tepe, K.: A high performance cluster-based broadcasting algorithm for wireless ad hoc networks based on a novel gateway selection approach. In: PE-WASUN '05, ACM (2005) 65–70
11. Peng, W., Lu, X.: Ahbp: An efficient broadcast protocol for mobile ad hoc networks. Journal of Computer Science and Technology **16**(2) (2001)
12. Lou, W., Wu, J.: Double-covered broadcast (dcb): a simple reliable broadcast algorithm in manets. In: INFOCOM 2004. Volume 3.
13. et al., P.J.: Performance analysis of OLSR multipoint relay flooding in two ad hoc wireless network models. RSRCP, Special issue on Mobility and Internet (2001)
14. Wu, J., Dai, F., Gao, M., Stojmenovic, I.: On calculating power-aware connected dominating sets for efficient routing in ad hoc wireless networks. IEEE/KICS Journal of Communications and Networks **4** (2002) 59–70
15. Blum, J., Ding, M., Thaeler, A., Cheng, X.: Connected dominating set in sensor networks and manets. In: Handbook of Combinatorial Optimization, D.-Z. Du and P. Pardalos, Kluwer Academic Publisher (2004) 329–369
16. Leclerc, T., Ciarletta, L., Andronache, A., Rothkugel, S.: Olsr and wcpd as basis for service discovery in manets. In: UBICOMM '08, IEEE (2008)
17. Gorgen, D., Frey, H., Hiedels, C.: Jane-the java ad hoc network development environment. In: ANSS '07, IEEE Computer Society (2007)
18. Blažević, L., Giordano, S., Le Boudec, J.Y.: Self organized terminode routing. Cluster Computing **5**(2) (2002)
19. Zeroconf: Zerconf. http://www.zeroconf.org/